UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/720,262	11/25/2003	Bjorn-Harald Sjogren	115719	4105

29078 7590 08/20/2007

CHRISTIAN D. ABEL
ONSAGERS AS
POSTBOKS 6963 ST. OLAVS PLASS
NORWAY, N-0130
NORWAY

EXAMINER	
DWIVEDI, MAHESH H	
ART UNIT	**PAPER NUMBER**
2168	

MAIL DATE	DELIVERY MODE
08/20/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/720,262	SJOGREN ET AL.	
	Examiner	Art Unit	
	Mahesh H. Dwivedi	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 May 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *31-45* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *31-45* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 November 2003* is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 10/02/2006 has been entered.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 31-34, 36-39, and 41-44 are rejected under 35 U.S.C. 102(b) as being anticipated by **Jacobs et al.** (U.S. Patent 6,105,025).

4. Regarding claims 31 and 41, **Jacobs** teaches a constraint enforcer and database system comprising:

A) a set of constraints that governs the integrity of information stored in the data system (Column 7, lines 1-9, 43-64);

B) said enforcer being arranged to delay constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction (Column 7, lines 1-9, 43-64, Column 9, lines 17-29, Column 10, lines 11-19);

C) the constraint enforcer comprising: a stack maker module, arranged for creating and updating said check stack (Column 9, lines 17-29, Column 10, lines 13-31);

D) said stack maker module being operatively connected to a runtime module in the database system (Column 6, lines 48-51, Column 9, lines 17-29); and

E) arranged to receive data from said runtime module (Column 6, lines 48-51, Column 9, lines 17-29);

F) an enforcer module (Column 7, lines 1-9, 43-64);
G) arranged to receive check data from the check stack (Column 7, lines 1-9, 43-64);
H) to process the check data received from the check stack (Column 7, lines 1-9, 43-64); and
I) to provide resulting data to the runtime module (Column 6, lines 48-51, Column 9, lines 17-29);
J) a conceptual rules module wherein said constraints are stored in the form of rules for prescribing permitted states and transitions that the database can undertake (Column 2, lines 65-67-Column 3, lines 1-16, Column 7, lines 1-6, Column 10, lines 15-17);
K) the conceptual rules module being operatively connected to said stack maker (Column 6, lines 48-51, Column 9, lines 17-29);
L) said stack maker module is arranged to retrieve constraints from said conceptual rules module (Column 7, lines 1-6, Column 9, lines 17-29, and Column 10, lines 13-31).

The examiner notes that **Jacobs** teaches **"a set of constraints that governs the integrity of information stored in the data system"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6), "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19), and "Uniqueness required index 308 is a B-tree structured index created on column 302N" (Column 10, lines 15-17). The examiner further notes that **Jacobs** teaches **"said enforcer being arranged to delay constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction"** as enforcement may be deferred until processing is completed for either a statement or transaction" (Column 7, lines 46-47), "constraint enforcement is deferred until the end of transaction (i.e., transaction level enforcement" (Column 7, lines 57-58). The examiner further notes that **Jacobs** teaches **"the constraint enforcer comprising: a stack maker module, arranged for creating and updating said check stack"** as "a list is generated for each uniqueness-required index

for each session" (Column 9, lines 17-19), "insert, delete, and update operations" (Column 10, line 13), and "Uniqueness required index 308 is a B-tree structured index created on column 302N" (Column 10, lines 15-17). The examiner further notes that **Jacobs** teaches **"said stack maker module being operatively connected to a runtime module in the database system"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that **Jacobs** teaches **"arranged to receive data from said runtime module"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that **Jacobs** teaches **"an enforcer module"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation" (Column 7, lines 1-3). The examiner further notes that **Jacobs** teaches **"arranged to receive check data from the check stack"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6). The examiner further notes that **Jacobs** teaches **"to process the check data received from the check stack"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation" (Column 7, lines 1-3) and "if constraint enforcement is deferred until the end of a transaction (i.e., transaction level enforcement)...are examined as an initial part of transaction commit processing" (Column 7, lines 57-62). The examiner further notes that **Jacobs** teaches **"to provide resulting data to the runtime module"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that it common knowledge that during a session, information and data is exchanged between application and database programs. The

examiner further notes that **Jacobs** teaches **"a conceptual rules module wherein said constraints are stored in the form of rules for prescribing permitted states and transitions that the database can undertake"** as "Uniqueness-required index 308 is a B-tree structured index created on column 302N" (Column 10, lines 15-17), and "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6). The examiner further notes that **Jacobs** teaches **"the conceptual rules module being operatively connected to said stack maker"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that it common knowledge that during a session, information and data is exchanged between application and database programs. The examiner further notes that **Jacobs** teaches **"said stack maker module is arranged to retrieve constraints from said conceptual rules module"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6). The examiner further notes that it common knowledge that during a session, information and data is exchanged between application and database programs.

Regarding claims 32, 37, and 42 **Jacobs** further teaches a constraint enforcer, method, and database system comprising:

A) wherein said check stack is stored on persistent or volatile memory. (Column 10, lines 13-31, Column 16, lines 51-53, Figure 6).

The examiner notes that **Jacobs** teaches **"wherein said check stack is stored on persistent or volatile memory"** as "Uniqueness-required index 308 is a B-tree structured index created on column 302N" (Column 10, lines 15-17) and "The present

invention can be implemented on a general purpose computer such as illustrated in FIG. 6...FIG. 6 also includes a video memory 614, main memory 615 and mass storage 612, all coupled to bi-directional system bus 618" (Column 16, lines 46-53).

Regarding claims 33, 38, and 43 **Jacobs** further teaches a constraint enforcer, method, and database system comprising:

A) wherein said stack maker module is further arranged to handle a modify operator as a delete operator followed by an insert operator (Column 10, lines 11-17, Figures 3A-3B).

The examiner notes that **Jacobs** teaches **"wherein said stack maker module is further arranged to handle a modify operator as a delete operator followed by an insert operator"** as "Examples of insert, delete, and update operations using a uniqueness-required index are provided with reference to FIGS. 3A-3B" (Column 10, lines 13-15).

Regarding claims 34, 39, and 44 **Jacobs** further teaches a constraint enforcer, method, and database system comprising:

A) wherein said constraints are transaction constraints (Abstract).

The examiner notes that **Jacobs** teaches **"wherein said stack maker module is further arranged to handle a modify operator as a delete operator followed by an insert operator"** as "Constraints can be enforced at the end of the processing of a statement, a transaction, or within a transaction at a savepoint" (Abstract).

Regarding claim 36, **Jacobs** teaches a method comprising:

A) a set of constraints that governs the integrity of information stored in the data system (Column 7, lines 1-9, 43-64);

B) the constraints being stored in a conceptual rules module in the form of rules for prescribing permitted states and transitions that the database can undertake (Column 2, lines 65-67-Column 3, lines 1-16, Column 7, lines 1-6, Column 10, lines 15-17);

C) said method comprising the steps of: delaying constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction (Column 7, lines 1-9, 43-64, Column 9, lines 17-29, Column 10, lines 11-19);
D) by a stack maker module operatively connected to a runtime module in said database system (Column 6, lines 48-51, Column 9, lines 17-29);
E) receiving data from said runtime module (Column 6, lines 48-51, Column 9, lines 17-29);
F) creating and updating said check stack (Column 9, lines 17-29); and
G) retrieving constraints from said conceptual rules module (Column 7, lines 1-6, Column 9, lines 17-29, and Column 10, lines 13-31); and
H) by an enforcer module: receiving check data from the check stack (Column 7, lines 1-9, 43-64);
I) processing the check data received from the check stack (Column 7, lines 1-9, 43-64); and; and
J) providing resulting data to the runtime module (Column 6, lines 48-51, Column 9, lines 17-29);.

The examiner notes that **Jacobs** teaches **"a set of constraints that governs the integrity of information stored in the data system"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6), "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19), and "Uniqueness required index 308 is a B-tree structured index created on column 302N" (Column 10, lines 15-17). The examiner further notes that **Jacobs** teaches **"the constraints being stored in a conceptual rules module in the form of rules for prescribing permitted states and transitions that the database can undertake"** as "Uniqueness-required index 308 is a B-tree structured index created on column 302N" (Column 10, lines 15-17), and "When a data value is created or added during a transaction or statement, a check is

made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6). The examiner further notes that **Jacobs** teaches **"said method comprising the steps of: delaying constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction"** as enforcement may be deferred until processing is completed for either a statement or transaction" (Column 7, lines 46-47), "constraint enforcement is deferred until the end of transaction (i.e., transaction level enforcement" (Column 7, lines 57-58). The examiner further notes that **Jacobs** teaches **"by a stack maker module operatively connected to a runtime module in said database system"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that **Jacobs** teaches **"receiving data from said runtime module"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that it common knowledge that during a session, information and data is exchanged between application and database programs. The examiner further notes that **Jacobs** teaches **"creating and updating said check stack"** as "According to one embodiment of the invention, a list is generated for each uniqueness-required index for each session. Every time the counter of an index changes, an entry is added to the list associated with the index. The entry includes the new value of the non-uniqueness count of the index and a savepoint indicator. When a transaction is rolled back to a particular savepoint, entries are removed from the tail of the list until an entry is encountered that has a savepoint indicator that identifies the savepoint to which the transaction is being rolled-back. The entry that contains the corresponding savepoint-indicator will indicate the uniqueness count value that the corresponding index should have after the roll-back is performed" (Column 9, lines 17-29). The examiner further notes that **Jacobs** teaches **"retrieving constraints from**

said conceptual rules module" as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6). The examiner further notes that it common knowledge that during a session, information and data is exchanged between application and database programs. The examiner further notes that **Jacobs** teaches **"by an enforcer module: receiving check data from the check stack"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6). The examiner further notes that **Jacobs** teaches **"processing the check data received from the check stack"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation" (Column 7, lines 1-3) and "if constraint enforcement is deferred until the end of a transaction (i.e., transaction level enforcement)...are examined as an initial part of transaction commit processing" (Column 7, lines 57-62). The examiner further notes that **Jacobs** teaches **"providing resulting data to the runtime module"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that it common knowledge that during a session, information and data is exchanged between application and database programs.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

7. Claims 35, 40, and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Jacobs et al.** (U.S. Patent 6,105,025) as applied to claims 31-34, 36-39, and 41-44 above, and further in view of **Jenkins** (U.S. Patent 5,899,993).

8. Regarding claims 35, 40, and 45, **Jacobs** does not explicitly teach a constraint enforcer, method, and database system comprising:

A) wherein said constraints are selected from: primary keys, foreign keys, subset constraints, and exclude constraints.

Jenkins, however, teaches **"wherein said constraints are selected from: primary keys, foreign keys, subset constraints, and exclude constraints"** as "Not null, check, unique key, primary key, and foreign key constraints are defined in terms of queries on the database which are required to return an empty result. A constraint can be validated by running the constraint's defining query and confirming that the result is empty. The actual validation procedure may vary from implementation to implementation. For example, the validation procedure may be different for database systems that do not require processes to obtain a shared lock before reading data than for database systems that require processes to obtain shared locks before reading data" (Column 7, lines 10-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Jenkins's** would have allowed **Jacobs's** to provide a method for allowing a variety of constraints to be enabled when database operations occur, as noted by **Jenkins** (Column 3, lines 30-34).

Response to Arguments

9. Applicant's arguments filed on 05/24/2007 have been fully considered but they are not persuasive.

Applicant argues on page 2, that **"Jacobs thus teaches that a non-uniqueness count, i.e., an incremented count of violations of the uniqueness constraint, is stored in memory. Clearly, Jacobs does not disclose that constraints are stored in a conceptual rules module in the form of rules for prescribing permitted states and transitions that the database can undertake"**. However, the examiner wishes to point to Columns 2-3 and 6 of **Jacobs** which state "It is sometimes desirable to put limitations on the values that are stored in a column. For example, it may be desirable to prohibit duplicate values in a column that represents the social security numbers for employees. A prohibition against duplicate values within a column is referred to as a uniqueness constraint. Stated generally, a uniqueness constraint prohibits two or more rows of a table from having the same value in a column or a group of columns. When two rows of a table have the same value in a column or a group of columns, a "uniqueness constraint violation" is said to have occurred" (Column 2, lines 66-67-Column 3, lines 1-9) and "Instead of using a unique index with a separate workspace to store duplicate values, the present invention uses an index that is capable of containing duplicate occurrences of a value (a "uniqueness-required index"). When there is more than one occurrence of a value, the duplicate occurrence is stored in the uniqueness-required index itself, eliminating the need for a separate workspace. The number of uniqueness constraint violations in each uniqueness-required index are tracked by using a "non-uniqueness" count... The present invention is discussed with reference to the enforcement of uniqueness constraints on a column or columns of a table in a DBMS. However, the present invention can be used to enforce uniqueness in other contexts. For example, the present invention can be used to manage queues, or to schedule transactions with a database system. Applications that allocate resources can use the present invention to ensure that multiple allocations or assignments of the same resource are not made. Examples of such applications include airline seat reservation and hotel room assignments" (Column 6, lines 8-17, lines 25-35). The examiner further

wishes to state that the uniqueness constraints of **Jacobs** are rules (see definition of uniqueness constraint: "A prohibition against duplicate values within a column is referred to as a uniqueness constraint"). The examiner further wishes to state that the uniqueness constraints prohibits certain actions on a database, and as a result, is enforcing a predefined rule.

Applicant argues on page 3, that **"Jacobs does not disclose the step of executing entries on the check stack at the end of the transaction. In Jacobs, the memory or list of constraints contains counts of uniqueness constraint violations"**. However, the examiner wishes to point to Column 7 of **Jacobs** which state "Enforcement of uniqueness constraints are deferred to the appropriate enforcement time. In a DBMS, for example, enforcement may be deferred until processing is completed for either a statement or a transaction. If constraint enforcement is deferred until the end of the statement (i.e., statement level enforcement), the non-uniqueness counts associated with the uniqueness-required indexes referenced during the processing of the statement are examined during the final phase of execution of the statement processing. If any of the uniqueness-required indexes are "currently non-unique", the effects of the statement are reversed (i.e., not made permanent in the database). Further, an error indication can be raised, or logged. Similarly, if constraint enforcement is deferred until the end of a transaction (i.e., transaction level enforcement), the non-uniqueness counts associated with the uniqueness-required indexes referenced during the processing of the transaction are examined as an initial part of transaction commit processing. If any of the uniqueness-required indexes are "currently non-unique", the commit is not performed. Further, an error indication is raised, or logged" (Column 7, lines 49-64). The examiner further wishes to state that it is clear that the constraint determination of **Jacobs** occurs at the end of the transaction processing, and that the uniqueness constraints prohibits certain actions on a database, and as a result, is enforcing a predefined rule/check.

Applicant argues on page 3, that **"Since Jacobs does not disclose a conceptual rules module wherein constraints in the form of rules are stored, Jacobs evidently does not disclose the retrieving of constraints form a**

conceptual rules module either". However, the examiner wishes to point to Columns 2-3 and 6 of **Jacobs** which state "It is sometimes desirable to put limitations on the values that are stored in a column. For example, it may be desirable to prohibit duplicate values in a column that represents the social security numbers for employees. A prohibition against duplicate values within a column is referred to as a uniqueness constraint. Stated generally, a uniqueness constraint prohibits two or more rows of a table from having the same value in a column or a group of columns. When two rows of a table have the same value in a column or a group of columns, a "uniqueness constraint violation" is said to have occurred" (Column 2, lines 66-67-Column 3, lines 1-9) and "Instead of using a unique index with a separate workspace to store duplicate values, the present invention uses an index that is capable of containing duplicate occurrences of a value (a "uniqueness-required index"). When there is more than one occurrence of a value, the duplicate occurrence is stored in the uniqueness-required index itself, eliminating the need for a separate workspace. The number of uniqueness constraint violations in each uniqueness-required index are tracked by using a "non-uniqueness" count... The present invention is discussed with reference to the enforcement of uniqueness constraints on a column or columns of a table in a DBMS. However, the present invention can be used to enforce uniqueness in other contexts. For example, the present invention can be used to manage queues, or to schedule transactions with a database system. Applications that allocate resources can use the present invention to ensure that multiple allocations or assignments of the same resource are not made. Examples of such applications include airline seat reservation and hotel room assignments" (Column 6, lines 8-17, lines 25-35). The examiner further wishes to state that the uniqueness constraints of **Jacobs** are rules (see definition of uniqueness constraint: "A prohibition against duplicate values within a column is referred to as a uniqueness constraint"). The examiner further wishes to state that the uniqueness constraints prohibits certain actions on a database, and as a result, is enforcing a predefined rule.

Applicant argues on page 3, that **"The dependent claim 39 specifies constraints as transaction constraints, thus further distinguishing the claim form**

the disclosure of Jacobs, which merely mentions the enforcement of uniqueness constraints". However, the examiner wishes to point to the Abstract and Column 7 of **Jacobs** which state "Constraints can be enforced at the end of the processing of a statement, a transaction, or within a transaction at a savepoint" (Abstract) and "Similarly, if constraint enforcement is deferred until the end of a transaction (i.e., transaction level enforcement), the non-uniqueness counts associated with the uniqueness-required indexes referenced during the processing of the transaction are examined as an initial part of transaction commit processing. If any of the uniqueness-required indexes are "currently non-unique", the commit is not performed. Further, an error indication is raised, or logged" (Column 7, lines 57-64). The examiner further wishes to state that the uniqueness constraints of **Jacobs** are rules (see definition of uniqueness constraint: "A prohibition against duplicate values within a column is referred to as a uniqueness constraint"). The examiner further wishes to state that it is clear that **Jacobs** clearly teaches constraints on transactions.

Applicant argues on page 3, that **"The dependent claim 40 specifies the constraints to be selected from/...of uniqueness constraints"**. Applicant's arguments with respect to claim 40 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent 5,706,494 issued to **Cochrane et al.** on 06 January 1998. The subject matter disclosed therein is pertinent to that of claims 31-45 (e.g., methods issue and enforce constraints on database systems).

U.S. Patent 6,453,314 issued to **Chan et al.** on 17 September 2002. The subject matter disclosed therein is pertinent to that of claims 31-45 (e.g., methods issue and enforce constraints on database systems).

U.S. Patent 5,408,657 issued to **Bigelow et al.** on 18 April 1995. The subject matter disclosed therein is pertinent to that of claims 31-45 (e.g., methods issue and enforce constraints on database systems).

Contact Information

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mahesh Dwivedi whose telephone number is (571) 272-2731. The examiner can normally be reached on Monday to Friday 8:20 am – 4:40 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached (571) 272-3642. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mahesh Dwivedi
Patent Examiner
Art Unit 2168

August 13, 2007

TIM VO
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100